Cloud-powered Wearable Fitness Tracking

Vishnu Vardhini
University Of Edinburgh
UK

ABSTRACT

The paper emphasises the implementation of the end to end prototype of a wearable fitness tracking device which is a IoT based Human Activity Recognition(HAR), to measure the physiological variables using the sensors in the HAR system, STMicroelectronics B-L475E-IOT01A1 Discovery IoT node. The kit has a built-in accelerometer and gyrometer for posture and activity analysis. The data captured by the HAR system is then transmitted over the wifi to the Pelion Management System. Since the whole idea is to recognize the activity from the data, a machine learning model is developed. The sensor data along with the predictions made by the trained model are stored in the database deployed to the cloud server from where the local visualization of the HAR is achieved by making API calls to the cloud server.

KEYWORDS

IoT, Embedded, Fitness Tracker, HAR, Random Forest Classifier, IOT cloud, security

1 INTRODUCTION

Internet of Things(IoT) is a popularly emerging paradigm, with respect to fitness tracking IoT enables the monitoring of an object using sensors, which can be connected to the internet for seamless availability of data with reduced latency and to perform a real-time data analysis [2] [8].

In the IoT based Human Activity Recognition(HAR), There are several arenas where HAR has been researched, evaluated and developed to the point that several commonly available products have HAR systems built in [2]. A classical example would be the fitness tracking devices such as bands in which the HAR aids in real time analysis of day to day and sleep activity [11].

The HAR uses several mechanisms to recognise the activities with the help of built in sensors to recognise the positions and movements [11]. Although there are a handful of commercial products using these systems, there are few setbacks relating to the data accuracy, coverage and cost [2]. Hence the proposed prototype will try to minimise the setback to an extent using the IoT component as a wearable over a period of time to capture the activity or the movement with reduced latency.

The prototype model is designed to perform the personal tracking. It can also be extended to apply in Healthcare services, provided the patients or users who use it are not in critical condition. An ideal scenario can be, remote patient monitoring where the practitioner would want to monitor the physical activities of the user.

2 DESIGN OVERVIEW

The Design overview section will discuss the key design choices made inorder to design the architecture of the prototype.

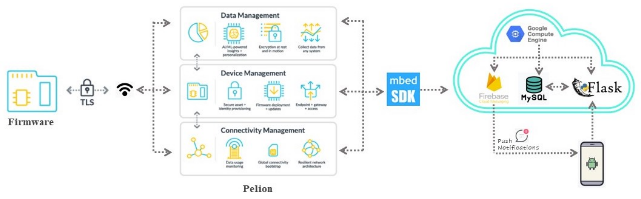

Firmware
TLS
Data Management
AI/ML-powered insights + personalization
Encryption at rest and in motion
Collect data from any system
Device Management
Secure asset + identity provisioning
Firmware deployment + updates
Endpoint + gateway + access
Connectivity Management
Data usage monitoring
Global connectivity bootstrap
Resilient network architecture
Pelion
mbed SDK
Google Compute Engine
Firebase
MySQL
Flask
Push Notifications

Figure 1: System Architecture

2.1 Firmware Integration

The Embedded Application to read the relevant sensor readings for the fitness tracking apps was implemented by integrating the sensor interface of B-L475E-IOT01A1 Discovery IoT node and the Mbed api. The device comprises LSM6DSL which is a system-in-package of a 3-axis accelerometer and 3-axis gyroscope. The features of LSM6DSL include various posture and motion detection along with step counter and time stamping which can be very essential for a fitness tracking app.

Inorder to initialize the accelerometer and the gyroscope sensors,respective sensors are invoked. After the initialization of the sensors the respective readings were retrieved. The 3-axis accelerometer readings and the 3-axis gyroscope readings readings are read using the The readings were concatenated together and stored as a single string. This choice was made to reduce the latency in send the readings to Pelion Management System and to send them in one shot.

According to the device user manual[9] the LSM6DSL has a full-scale acceleration range of $2/4/8/16g$ and an angular rate range of $125/245/500/1000/2000dps$. Therefore the measurement unit of the readings for the accelerometer is g cm/s^2 and for gyroscope it is dps, degrees per second.

2.2 Communication With Pelion Management System

The Pelion Device Management platform is used as the service that can manage and interact with the connected device. The Pelion Management system ensures in securely onboarding the device and interacting with it. The major goal of the device management system is to provide end to end lifecycle management from factory provisioning by securely loading the device to efficiently monitor the device resources until the device is disembarked. [1] The B-L475E-IOT01A1 Discovery IoT node has an onboard 802.11 b/g/n compliant Wi-Fi module which is capable of transmitting sensor readings to the pelion device management platform using the lightweight machine-to-machine (LwM2M) protocol.

Access Management: To begin with it is necessary to undergo an authentication mechanism which requires setting up an mbed account and setting up an account management to manage users, application access keys and their access rights.

Device Provisioning: Next step was to set up the device for secured connectivity and communication using the Public key infrastructure (PKI) dependent on X.509 certificate to authenticate the server and the device using public key encryption.The Provisioning process is facilitated by the Device Management Client which securely stores the configuration data provided by the user and invokes the device management services when the device is ready to connect. Inorder to create that trust relationship it is required to install certificates on both Device Management and the device.

Connecting Device: The connection flow involves identifying the device name which is a unique ID generated by device management when the device connects for the first time and the endpoint name which is the name that is given to the device in the factory. Later connection establishment is achieved through bootstrapping which provides the LwM2M server account credentials for registration. Finally the device is registered by directly connecting to the LwM2M server using the credentials received during bootstrapping. The device is deregistered from the LwM2M server when they are no longer needed.

Managing Device: Once the Device was connected the resources can be managed and the communication with the device is achieved in an asynchronous way where the Device Management queues the operation and delivers the result of an operation, emitted as an event through the event notification channel. The resources are stored in a hierarchical data structure of device objects and resources. Where the objects are the collection of resources and any piece of information from the device is a resource the resource can be a static or a dynamic value. A resource can also be an array with multiple values.

Application Configuration: The firmware to transmit the sensor readings was built using the mbed studio, the default connectivity configuration is overridden by adding the wifi hotspot username and credentials later the network interface is initialised to connect to the WiFi.

The sensor readings which were obtained as single string is now assigned to a resource path as a M2M resource and are binded to a webpath. As the sensor data is in single string only one resourse path is used avoiding the need to create binding for multiple resources.

Since the prototype is still in testing phase the measurement sampling rate is determined to be to 5 seconds, therefore the readings are updated to the EventQueue every 5 seconds so no temporal information is lost. This also ensures that the latency is not too high and also updation is not too frequent which can lead to network congestion.

Api key and Certificate: Inorder to provide the privilege for the application to access the pelion device management system, the API key which needs to be created from the device management portal with developer access rights can be loaded by logging in to the portal from the mbed studio's pelion device management icon, this will also load the developer connect certificate to securely connect the device after successful secure login. The key can also be added directly by using the Pelion DM Account Settings from the settings option available by clicking on the pelion device management icon. Once the device is registered to the portal it can be listed under the Device directory -> Devices. By clicking on the registered device the sensor readings of the resources are available under their created webpath.

Firmware Update: The device management also provides support to update the firmware to deployed services by building a new image from the device that has a device management update client running on it which takes care of validating, installing and monitoring the firmware updates. To ensure that the updates are from a trusted source it validates the update using the Elliptic Curve Digital Signature Algorithm. The algorithm validates if the computed hash and the signature hash are the same to ensure the authenticity of the signer.

2.3 Machine Learning

The learning or training phase is the initial phase of any recognition algorithm. The program containing the algorithm and the associated relationship between data and activity acquired from learning is considered our Trained Model. To develop and pre train the model that can perform HAR such running or walking the Kaggale dataset [6] is used. This dataset contains previously collected accelerometer and gyroscope measurements. The purpose of using this was to have some basic pretrained model.

Preliminary Analysis

Before diving into developing the machine learning model it is important to first understand the dataset before making assumptions. It is clear from the documentation that the dataset was accumulated by Viktor Malyi consisting of 88588 sensor data samples of accelerometer and gyroscope from iPhone 5c in 10 seconds interval and $5.4/s$ frequency. The axis values of each sensor are represented in each column. It is also evident from the Apple documentation[4],[5] that the accelerometer values are denoted in G which is a unit of gravitation force equal to that exerted by the earth's gravitational field in $m/s2$ whereas the gyroscope values are measured in radians per second . Therefore the sensor readings acquired from the device was required to be converted to match the dataset measurement unit.

Best Fit Classifier

For simplicity and to obtain accurate predictions Random Forest Decision Tree classifier was used. The advantage of using it is their interpretability, the learnt rules used to classify the data can be easily followed. The classifier also limits the overfitting without introducing error due to bias. To begin with, the dataset is split into train and test.

For data preparation the dataset was normalised using the mean to ensure numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. Once the training is completed the accuracy was tested to be 0.9913459005907364 which means that the model is 99.13 percent accurate. Hence it was clear that Random forest Classifier works best for this dataset.

Dataset Creation Once the model was trained, the sensor reading from the firmware was planned to be converted according to the existing dataset. Despite converting the measurement unit of the raw data to match with the dataset it was tricky to find the right way to replicate the format. Hence without much ado the plan was changed to create a fresh dataset.

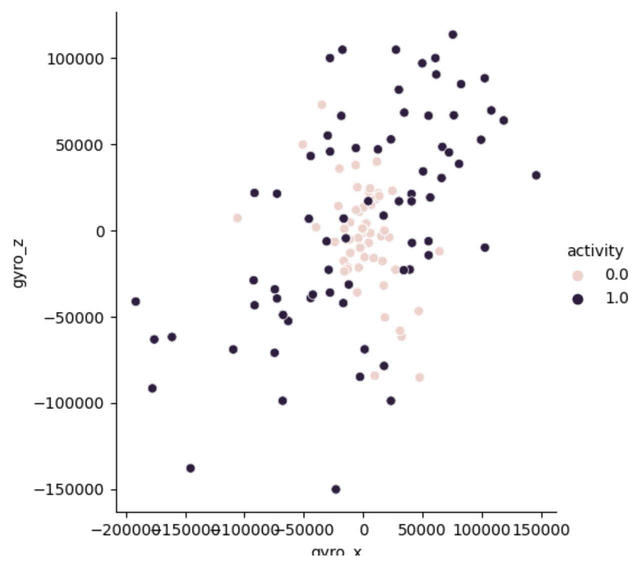


Figure 2: Data Distribution of Acceleration readings

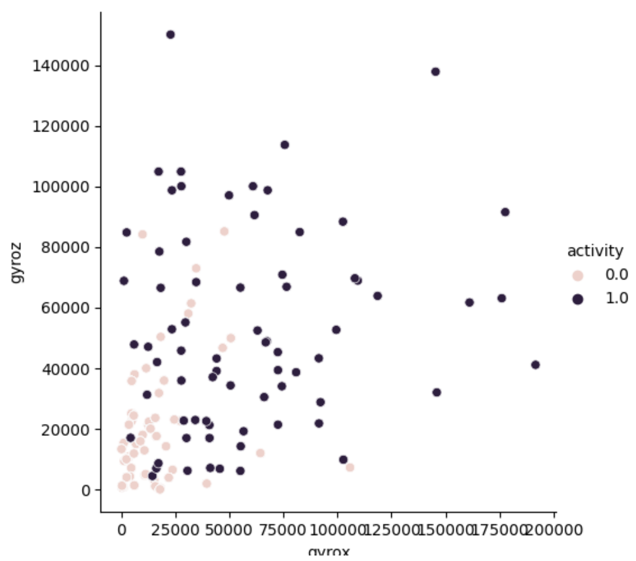


Figure 3: Data Distribution of Gyroscope readings

Using the device firmware a training dataset of 20,000 rows was created with equal walk and run distribution. Since the machine learning model skeleton was already in place the only change was to plugin the new dataset and pretrain it. The test accuracy was 100 percent and the training accuracy was 99 percent.

3 CLOUD

GCP was used as the cloud solution for this project: To store the predictions to the database, deploy the webservices and expose them as API's to interact with the predictions. In an IoT infrastructure cloud plays the vital role in remote visualisation, high availability, storage and scalability to name a few. However relying completely on managed services of one cloud infrastructure makes the architecture tightly coupled. If in case there is a requirement to migrate to another cloud infrastructure the IoT system's performance should not be compromised. Keeping that in mind loosely coupled cloud architecture was designed So that the components can be plugged out and plugged in to the new infrastructure with less hassles. For setting up the infrastructure the compute engine was spun up and necessary tools such as mbed cloud SDK, mysql, python and its dependencies were installed.

Backend Connectivity

Using the mbed cloud SDK for python the sensor readings were fetched by making call to the its connectAPI() to get the resource value from pelion management system as string stream of accelerometer and gyroscope readings and update them to the database.

For the backend mysql database was chosen due for its robustness and reliability. Not to mention mysql offers security to protect data including secure connections, authentication services, fine-grained authorization and controls, and data encryption.

The database is created with the schema replicating the walk/run kaggle dataset. The sensor readings are mapped to their corresponding rows. A utility file was created in python to open a secure connection to the database and compute over it. Later for storing the data's corresponding predictions, The trained model is imported as a pickle file which takes in the input readings and provides the predictions as "0"- walk and "1"- run.

Web API

The webservices was developed using Flask a popular framework for developing web applications. Flask is a lightweight WSGI web application framework. It is designed for quick and easy deployment of web services. Due to the simplicity of the prototype this lightweight framework was chosen. Additionally its flexibility to make changes and test on the fly was very helpful during the initial stages of the prototype development.

The API's were all developed as GET requests since the scope of the prototype is only to get real time predictions through the API's. Methods were created for getting past data, today's data and the current activity of the user by fetching the real time data from the database.

Firebase Cloud Messaging The next steps of the implementation involved in developing an android app to display few metrics and send periodic milestone alerts to the users. Though android has native push notification functionality, it gets complex to support connecting to the real time data available over the cloud. This is handled by the Firebase Cloud Messaging service, a cross-platform messaging solution that lets reliably send messages at no cost. Now that the real time predictions are being made setting up alerts based on the predictions were straightforward. The firebase establishes the secure connection to the android phone using API key generated while registering the application to ensure security. For testing purpose Push notification was set to be sent to the android phone after every one minute of walking.

4 MOBILE APPLICATION

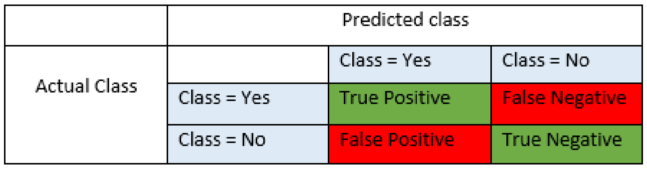
For Local Visualisation of the real time predictions an android application was developed. The application needs to be subscribed to the firebase console for receiving notifications. Since there is not much metrics targeted to be displayed only a single activity page is created. Firstly a layout was created and the logic to make call to the API was done, later the user interface was enhanced with bar chart to classify between walk and run data. A simple animation to the bar chart was also done.

Now the next step is to feed the UI with real time data. For making API calls from mobile to a webapp deployed on cloud server a HTTP request/ response architecture is a classical option. OkHttp is used as the HTTP client due to its support to handle android application's lifecycle. It also supports asynchronous calls and provides callback method to handle failure conditions without crashing the app when the webservice is down. The security provider for the Okhttp client is powered by BoringSSL which is the fork of OpenSSL that is designed to meet Google's needs [3].

Using the OkHttp webservice calls were made to fetch data during startup and the response is mapped to the UI and barchart animatiopn is started. The UI is refreshed with current activity every 15 seconds to reduce multiple calls to the webservice. Also For simplicity past data is only the yesterday's data.The UI displays todays data or yesterday's data along with its bar chart visulaisation when repective buttons are pressed.

5 EVALUATION

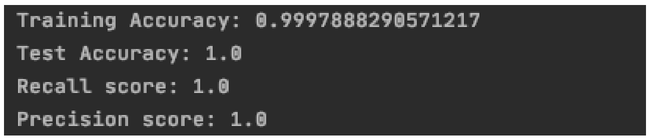
Evaluating the pretrained model is the most important task in the IoT project which delineates how good the predictions are made. The above metrics presented in the figure are calculated by the machine learning algorithm using confusion matrix where the True Positives (TP),False Negative (FN), False Positive (FP) and True Negative (TN) are plotted over the actual class and the predicted class [10].



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 4: Confusion Matrix

The Performance of the the prototype was estimated by evaluating the model for its accuracy, precision and recall scores computed by the Random forest classifier.



```
Training Accuracy: 0.9997888290571217
Test Accuracy: 1.0
Recall score: 1.0
Precision score: 1.0
```

Figure 5: Accuracy,Recall and Precision Scores

Accuracy is a intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. It is calculated as: $Accuracy = TP + TN/TP + FP + FN + TN$

The protype accuracy is really high due to the fact that we have a symmetric datasets where values of false positive and false negatives are almost same. Hence other mesures such as precision and recall was evaluated. $Precision = TP/TP{+}FP$ $Recall = TP/TP + FN$ Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. Whereas Recall is the ratio of correctly predicted positive observations to the all observations in actual class For this project the precision and recall was 100 percent which is very good.

However the dataset used is small with only of 20,000 rows. Hence the model needs to be updated with more data under various environmental and physiological conditions. Machine Learning is going to be an ongoing process and hence starting from a simpler dataset and enhancing it with additional data and parameters will keep things lightweight.

6 SECURITY

Though the security of the components are covered to an extent while discussing on the design of Pelion Management Sytem, cloud and the mobile app it is still not a fool proof system to mitigate security related vulnerabilities. There is always a pressing need to ensure the privacy of the users and minimise the issues related to security when building IOT systems.

It has been reported that fitness trackers are becoming the major target of deliberate attacks such as eavesdropping, unauthorized account access, IP address spoofing, fake firmware update, and so on Additionally as the Fitness Tracking devices enhance their functionalities wide range of sensitive data from the users are required to compute and visualise the real time predictions [7]. These data are always in stake of being hacked by malicious users.

With respect to this prototype GCP is the umbrella taking care of the infrastructure and is always in attention to the malicious users. Few identified vulnerabilities of this prototype is as follows:

- The entire database and web server is hosted on it.
- Also for firebase push notification the andorid device id is stored on the server.
- Yet another concern is making direct API calls on the server IP rather than making to their dns names.
- Proceeding with HTTP instead of creating certificates to make HTTPS.

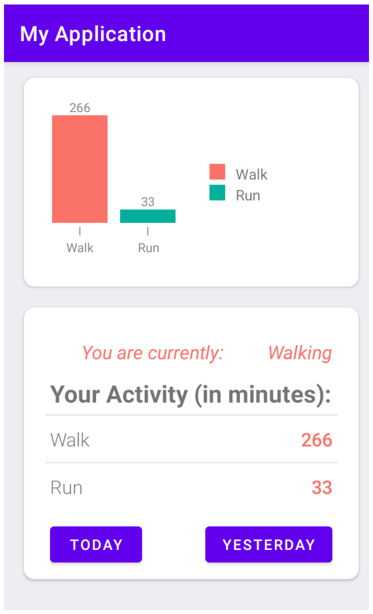
These are some major loopholes which can deteriorate the system and hence it is essential to take care of its security.When using GCP platform as the infrastructure google documentation advises few best practices. which I aim to incorporate in the future.

- Always keep in mind to design secure solutions from the start by following: separation of concerns, principle of least privilege, and regular audits.
- Simplify cloud governance using organizational policies and folders.
- Always segregate the people according to the roles and restrict unwanted access by leveraging IAM roles, Identity-Aware Proxy, and Identity Platform.

- Manage the access and authorization of resources by machines using service accounts.
- Secure the network with private IPs, firewalls, and Private Google Access.
- Mitigate DDoS attacks by leveraging Cloud DNS

7 PROTOTYPE EVIDENCE

The evidence of the prototype is presented as the screenshots of the mobile application which is been populated from the API's real time data

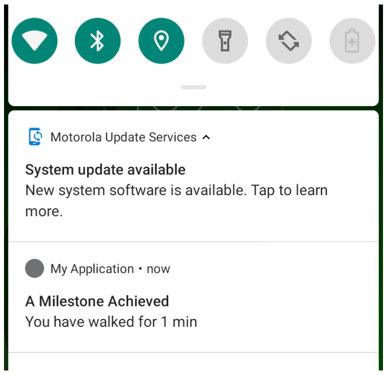


You are currently: Walking

Your Activity (in minutes):

Walk	266
Run	33

Figure 6: Homepage with Todays Activity


Motorola Update Services

System update available
New system software is available. Tap to learn more.

My Application • now

A Milestone Achieved
You have walked for 1 min

Figure 7: Push Notification from Firebase Cloud Messaging

8 FUTURE SCOPE

Despite the frequency of the data collection interval being 5 secs, the process of data being stored to the database and from there API calls posted by the android device to fetch data suffers latency issues. Inorder to overcome this edge processing needs can be implemented to perform few local processing at the mobile device rather than completely relying on the cloud.

Additionally only walk and run are considered in scope for the prototype which can be extended to multiple HAR such as posture, idle scenario such as sit/sleep and heart rate monitoring using the collected sensor readings many data processing can be achieved such as implementing the heart rate monitoring.

The IoT prototype is of an HAR-IoT system with only one user and might required to broadened by enchancing the dataset with data from varied set of people with different age, height or weight. When a large training data is used robustness of the system will increase [2]

Improved android application needs to be developed with the support to have offline sync of the data to reduce redundant hits to the web service. To handle the growing amount of real time data better visualisation metrics needs to be considered with the utilities to help the user identify the pattern of their activities, provide past data visualisation. Using these pattern another recognition model can be implemented to give lifestyle improvement suggestions but this comes with the cost of user's privacy and needs to be taken care of.

9 CONCLUSION

The implemented prototype is a base skeleton of an end to end HAR-IoT system which can recognise the activity performed by the user using the firmware, send the raw data to cloud for data processing, makes predictions using the machine learning model, stores the metrics and predictions to the mysql database hosted on GCP cloud server which exposes web API's using flask which performs database operations to fetch real time data when invoked by the android device. The android mobile app is built for local visualisation of real time predictions history of today/ yesterday's data, periodic auto refresh of current activity and to receive push notification when a small milestone is achieved. The prototype was evaluated with only few variables The future scope mentioned above needs to be considered to improve the overall performance of this prototype.

REFERENCES

[1] Copyright © Pelion 2021. 2019. *Pelion Device Management documentation.* Retrieved March 2, 2021 from https://developer.pelion.com/docs/device-management/current/welcome/index.html
[2] Diego Castro, William Coral, Camilo Rodriguez, Jose Cabra, and Julian Colorado. 2017. Wearable-Based Human Activity Recognition Using an IoT Approach. *Journal of Sensor and Actuator Networks* 6, 4 (2017). https://doi.org/10.3390/jsan6040028
[3] Inc. Copyright 2019 Square. 2019. *OkHttp.* Retrieved March 2, 2021 from https://square.github.io/okhttp/
[4] Copyright © 2021 Apple Inc. 2019. *Getting Raw Accelerometer Events.* Retrieved March 2, 2021 from https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events
[5] Copyright © 2021 Apple Inc. 2019. *Getting Raw Gyroscope Events.* Retrieved March 2, 2021 from https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events
[6] Viktor Malyi. 2017. *A dataset containing labeled sensor data from accelerometer and gyroscope.* Retrieved March 25, 2021 from https://www.kaggle.com/vmalyi/run-or-walk
[7] Florina Almenares Mendoza, Lucía Alonso, Andrés Marín López, and Daniel Díaz Sánchez Patricia Arias Cabarcos. 2018. Assessment of fitness tracker security: a case of study. In *Multidisciplinary Digital Publishing Institute Proceedings*, Vol. 2. 1235.

[8] Gina Roncancio, Mónica Espinosa, Manuel R Pérez, and Luis C Trujillo. 2017. Spectral sensing method in the radio cognitive context for IoT applications. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 756–761.

[9] STMicroelectronics. 2021. *iNEMO 6DoF inertial measurement unit (IMU), for smart phones and battery operated IoT, Gaming, Wearable and Consumer Electronics. Ultra-low power and high accuracy.* Retrieved March 25, 2021 from https://www.st.com/en/mems-and-sensors/lsm6dsl.html

[10] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther Van Der Knaap. 2011. Confusion Matrix-based Feature Selection. *MAICS* 710 (2011), 120–127.

[11] Ozgur Yurur, Chi Harold Liu, and Wilfrido Moreno. 2014. A survey of context-aware middleware designs for human activity recognition. *IEEE Communications Magazine* 52, 6 (2014), 24–31.